Enabling Scientific Workflows on FermiCloud using OpenNebula

Steven Timm
Grid & Cloud Services Department
Fermilab

Work supported by the U.S. Department of Energy under contract No. DE-AC02-07CH11359

Fermilab

Outline

Introduction—Fermilab and Scientific Computing

FermiCloud Project and Drivers

Applying Grid Lessons To Cloud

FermiCloud Project

Current and Future Interoperability

Reframing the Cloud Discussion

Fermilab and Scientific Computing

Fermi National Accelerator Laboratory:

- Lead United States particle physics laboratory
- ~60 PB of data on tape
- High Throughput Computing characterized by:
  - “Pleasingly parallel” tasks
  - High CPU instruction / Bytes IO ratio
  - But still lots of I/O. See Pfister: “In Search of Clusters”

Grid and Cloud Services Dept.

Operations:

Grid Authorization

Grid Accounting

Computing Elements

Batch Submission

All require high availability

All require multiple integration systems to test.

Also requires virtualization

And login as root

Solutions:

Development of authorization, accounting, and batch submission software

Packaging and integration

Requires development machines not used all the time

Plus environments that are easily reset

And login as root

HTC Virtualization Drivers

Large multi-core servers have evolved from 2 to 64 cores per box,

- A single “rogue” user/application can impact 63 other users/applications.
- Virtualization can provide mechanisms to securely isolate users/applications.

Typical “bare metal” hardware has significantly more performance than usually needed for a single-purpose server,

- Virtualization can provide mechanisms to harvest/utilize the remaining cycles.

Complicated software stacks are difficult to distribute on grid,

- Distribution of preconfigured virtual machines together with GlideinWMS and HTCondor can aid in addressing this problem.

Large demand for transient development/testing/integration work,

- Virtual machines are ideal for this work.

Science is increasingly turning to complex, multiphase workflows.

- Virtualization coupled with cloud can provide the ability to flexibly reconfigure hardware “on demand” to meet the changing needs of science.

Legacy code:

- Data and code preservation for recently-completed experiments at Fermilab Tevatron and elsewhere.

Burst Capacity:

- Systems are full all the time, need more cycles just before conferences.

FermiCloud – Initial Project Specifications

FermiCloud Project was established in 2009 with the goal of developing and establishing Scientific Cloud capabilities for the Fermilab Scientific Program,

- Building on the very successful FermiGrid program that supports the full Fermilab user community and makes significant contributions as members of the Open Science Grid Consortium.
- Reuse High Availabilty, AuthZ/AuthN, Virtualization from Grid

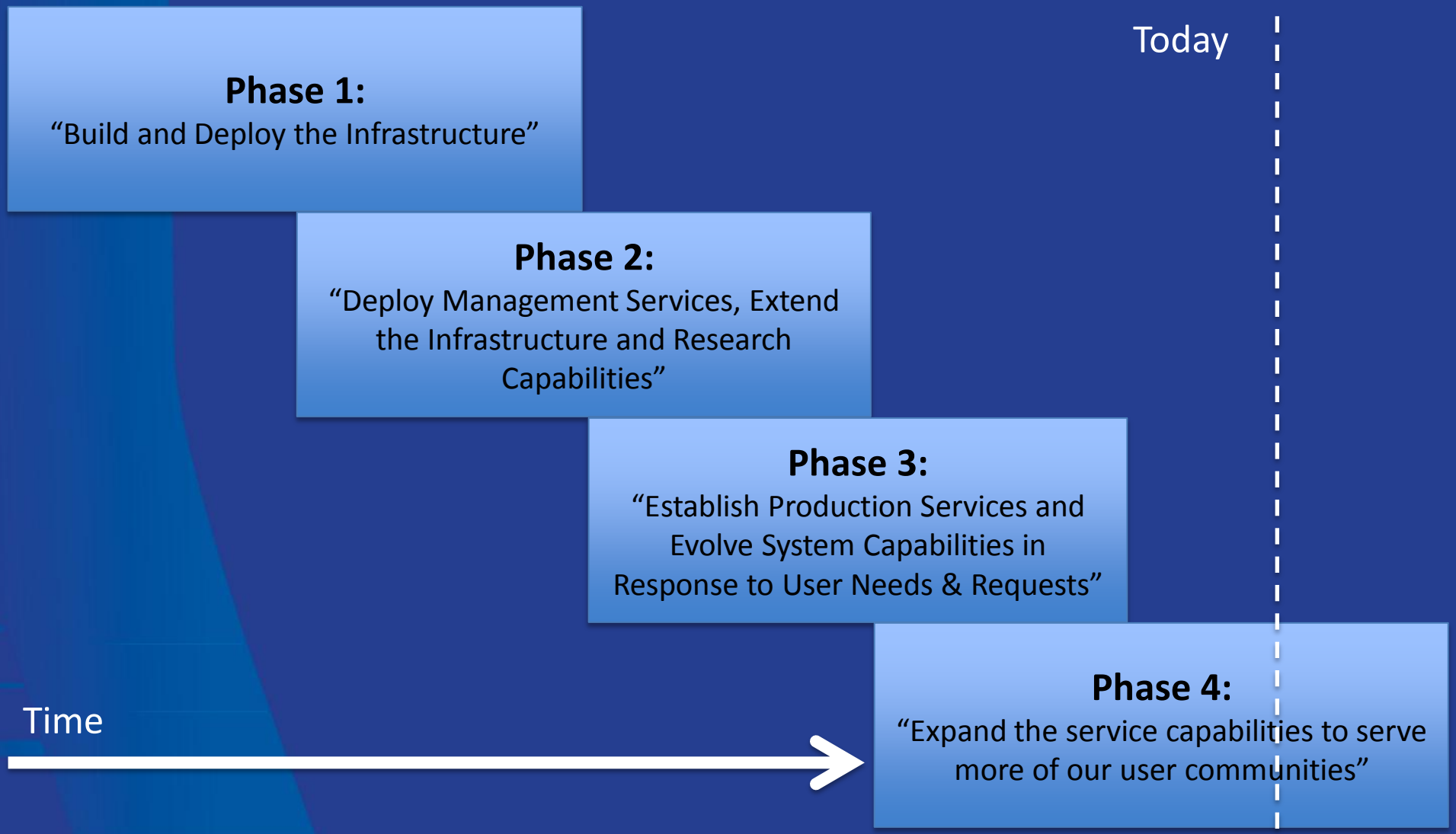
In a (very) broad brush, the mission of the FermiCloud project is:

- To deploy a production quality Infrastructure as a Service (IaaS) Cloud Computing capability in support of the Fermilab Scientific Program.
- To support additional IaaS, PaaS and SaaS Cloud Computing capabilities based on the FermiCloud infrastructure at Fermilab.

The FermiCloud project is a program of work that is split over several overlapping phases.

- Each phase builds on the capabilities delivered as part of the previous phases.

Overlapping Phases



Current FermiCloud Capabilities

The current FermiCloud hardware capabilities include:

- Public network access via the high performance Fermilab network,
  - This is a distributed, redundant network.
- Private 1 Gb/sec network,
  - This network is bridged across FCC and GCC on private fiber,
- High performance Infiniband network,
  - Currently split into two segments,
- Access to a high performance FibreChannel based SAN,
  - This SAN spans both buildings.
- Access to the high performance BlueArc based filesystems,
  - The BlueArc is located on FCC-2,
- Access to the Fermilab dCache and enStore services,
  - These services are split across FCC and GCC,
- Access to 100 Gbit Ethernet test bed in LCC (Integration nodes),
  - Intel 10 Gbit Ethernet converged network adapter X540-T1.

Typical Use Cases

Public net virtual machine:

- On Fermilab Network open to Internet,
- Can access dCache and Bluearc Mass Storage,
- Common home directory between multiple VM's.

Public/Private Cluster:

- One gateway VM on public/private net,
- Cluster of many VM's on private net.
- Data acquisition simulation

Storage VM:

- VM with large non-persistent storage,
- Use for large MySQL or Postgres databases, Lustre/Hadoop/Bestman/xRootd/dCache/OrangeFS/IRODS servers.

FermiGrid-HA2 Experience

In 2009, based on operational experience and plans for redevelopment of the FCC-1 computer room, the FermiGrid-HA2 project was established to split the set of FermiGrid services across computer rooms in two separate buildings (FCC-2 and GCC-B).

- This project was completed on 7-Jun-2011 (and tested by a building failure less than two hours later).
- FermiGrid-HA2 worked exactly as designed.

Our operational experience with FermiGrid-HA and FermiGrid-HA2 has shown the benefits of virtualization and service redundancy.

- Benefits to the user community – increased service reliability and uptime.
- Benefits to the service maintainers – flexible scheduling of maintenance and upgrade activities.

Experience with FermiGrid = Drivers for FermiCloud

Access to pools of resources using common interfaces:

- Monitoring, quotas, allocations, accounting, etc.

Opportunistic access:

- Users can use the common interfaces to “burst” to additional resources to meet their needs

Efficient operations:

- Deploy common services centrally

High availability services:

- Flexible and resilient operations

Additional Drivers for FermiCloud

Existing development and integration (AKA the FAPL cluster) facilities were:

- Technically obsolescent and unable to be used effectively to test and deploy the current generations of Grid middleware.
- The hardware was over 8 years old and was falling apart.
- The needs of the developers and service administrators in the Grid and Cloud Computing Department for reliable and “at scale” development and integration facilities were growing.
- Operational experience with FermiGrid had demonstrated that virtualization could be used to deliver production class services.

OpenNebula

- OpenNebula was picked as result of evaluation of Open source cloud management software.
- OpenNebula 2.0 pilot system in GCC available to users since November 2010.
- Began with 5 nodes, gradually expanded to 13 nodes.
- 4500 Virtual Machines run on pilot system in 3+ years.
- OpenNebula 3.2 production-quality system installed in FCC in June 2012 in advance of GCC total power outage—now comprises 18 nodes.
- Transition of virtual machines and users from ONe 2.0 pilot system to production system almost complete.
- In the meantime OpenNebula has done five more releases, will catch up shortly.

FermiCloud – Fault Tolerance

As we have learned from FermiGrid, having a distributed fault tolerant infrastructure is highly desirable for production operations.

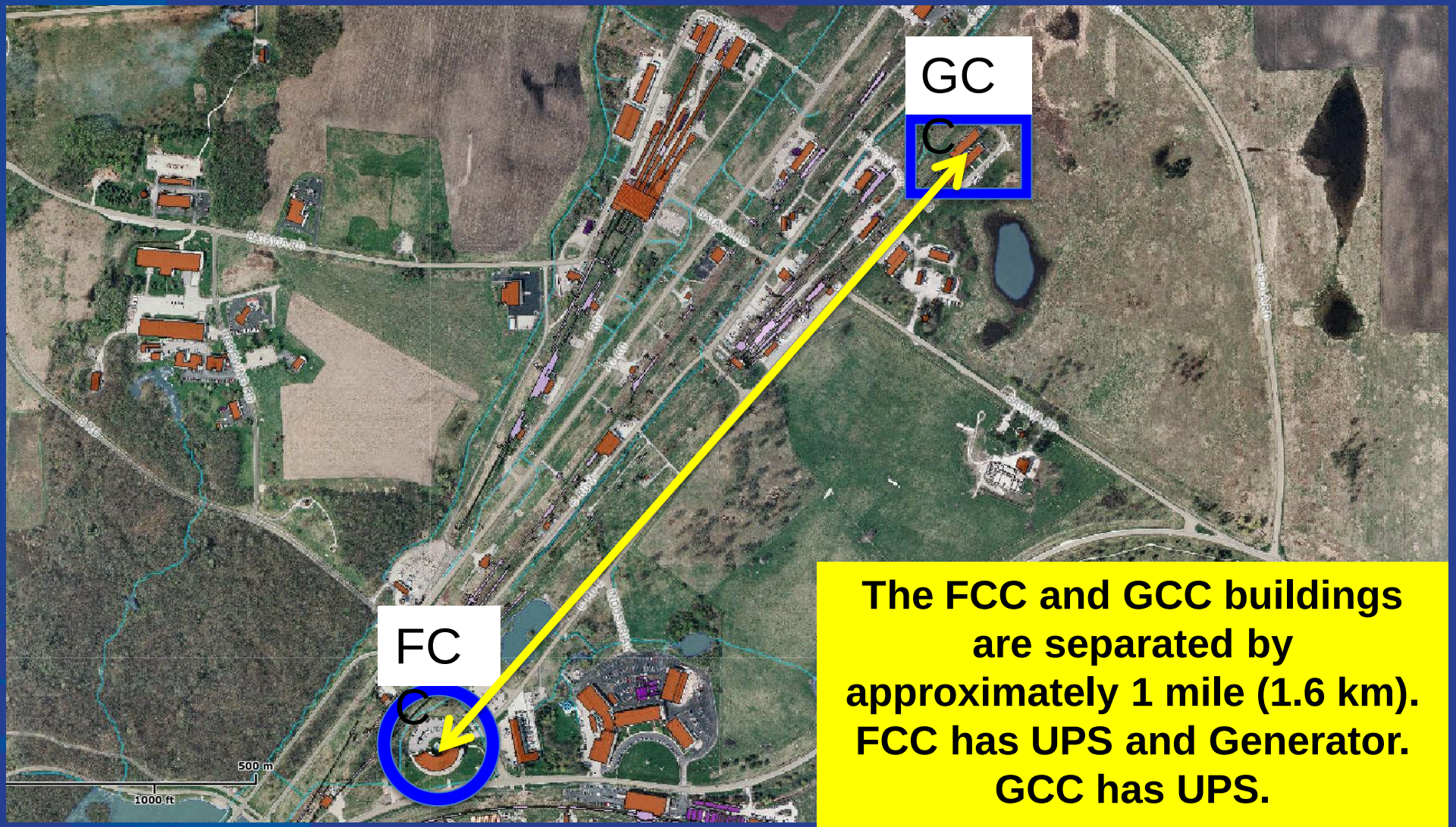
We are actively working on deploying the FermiCloud hardware resources in a fault tolerant infrastructure:

- The physical systems are split across two buildings,
- There is a fault tolerant network infrastructure in place that interconnects the two buildings,
- We have deployed SAN hardware in both buildings,
- We have a dual head-node configuration with HB for failover
- We have a GFS2 + CLVM for our multi-user filesystem and distributed SAN.
- SAN replicated between buildings using CLVM mirroring.

GOAL:

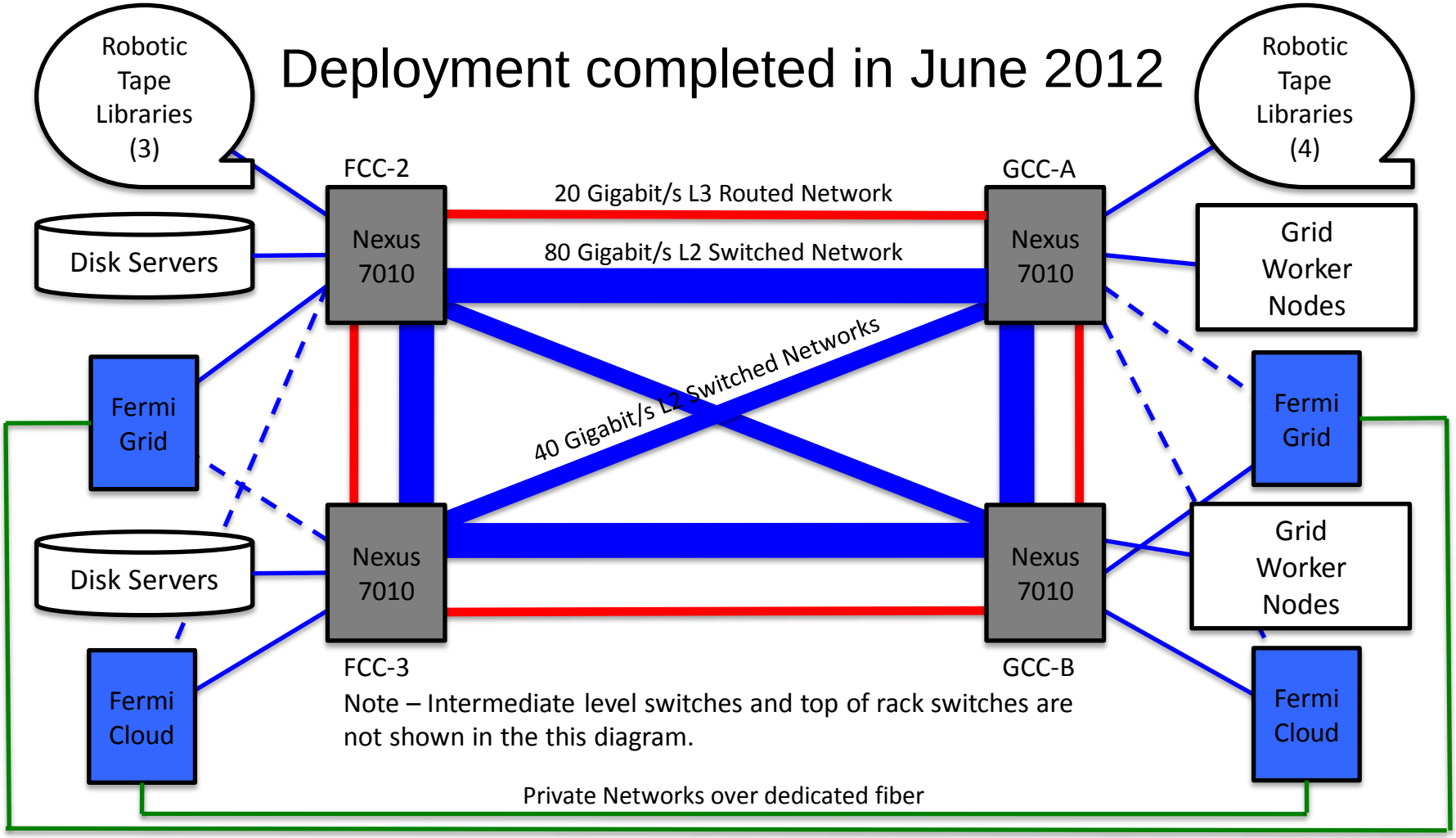
- If a building is “lost”, then automatically relaunch “24x7” VMs on surviving infrastructure, then relaunch “9x5” VMs if there is sufficient remaining capacity,
- Perform notification (via Service-Now) when exceptions are detected.

FCC and GCC



The FCC and GCC buildings are separated by approximately 1 mile (1.6 km). FCC has UPS and Generator. GCC has UPS.

Distributed Network Core Provides Redundant Connectivity



Distributed Shared File System

Design:

Dual-port FibreChannel HBA in each node,

Two Brocade SAN switches per rack,

Brocades linked rack-to-rack with dark fiber,

60TB Nexsan Satabeast in FCC-3 and GCC-B,

Redhat Clustering + CLVM + GFS2 used for file system,

Each VM image is a file in the GFS2 file system

LVM mirroring RAID 1 across buildings.

Benefits:

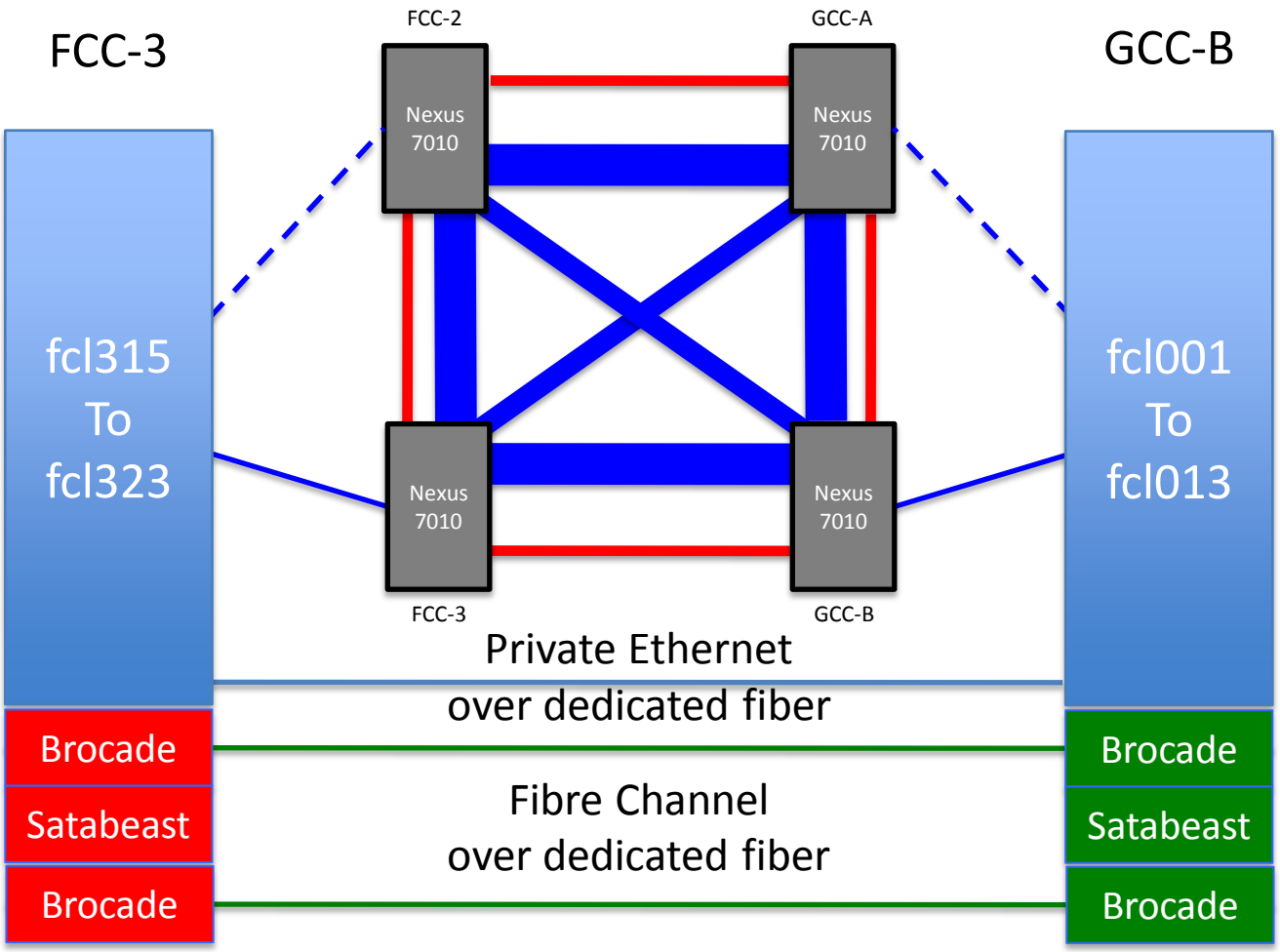
Fast Launch—almost immediate as compared to 3-4 minutes with ssh/scp,

Live Migration—Can move virtual machines from one host to another for scheduled maintenance, transparent to users,

Persistent data volumes—can move quickly with machines,

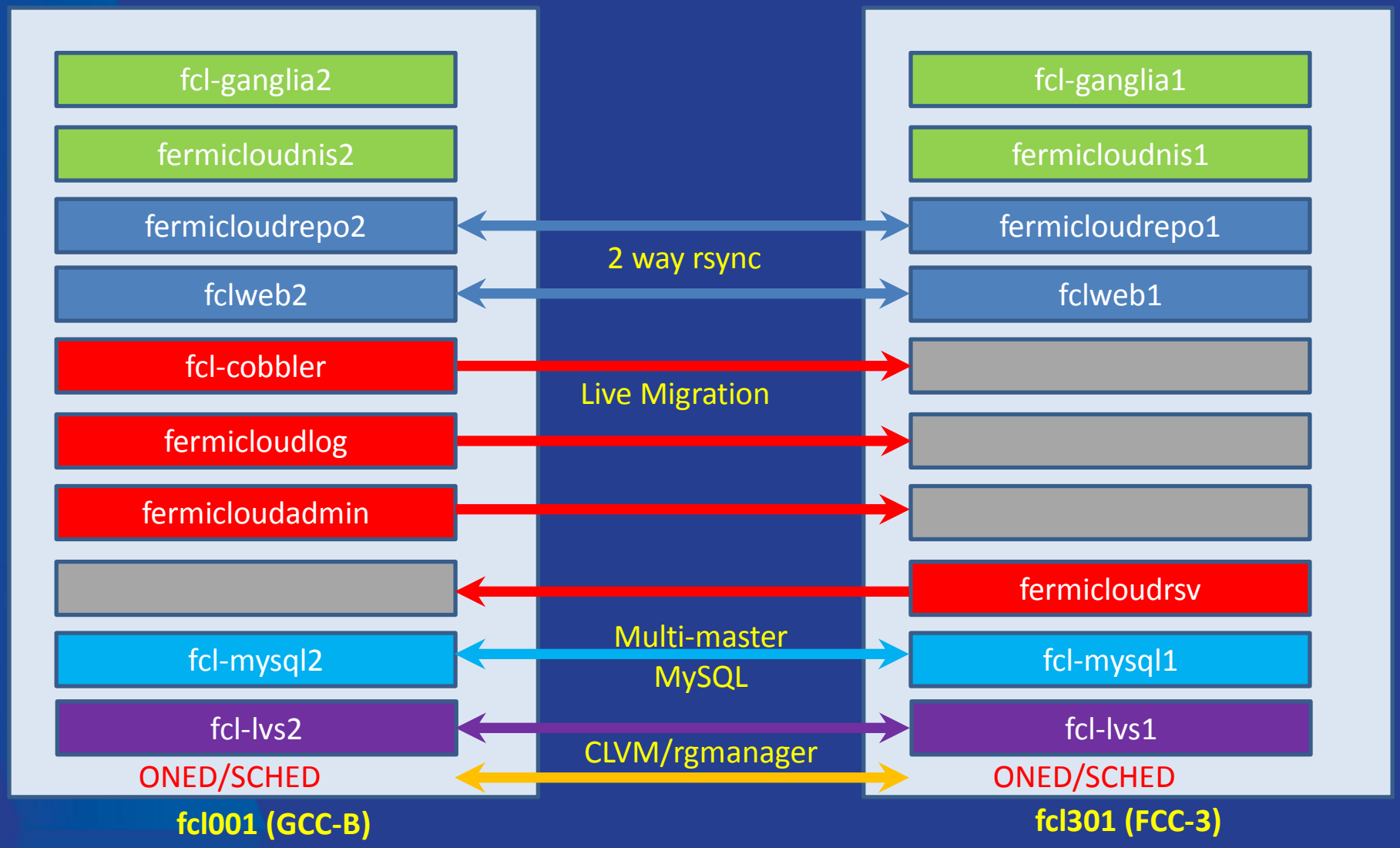
Can relaunch virtual machines in surviving building in case of building failure/outage,

FermiCloud – Network & SAN "Today"



FY2011 / FY2012

FermiCloud-HA Head Node Configuration



Cooperative R+D Agreement

Partners:

- Grid and Cloud Computing Dept. @FNAL
- Global Science Experimental Data hub Center @KISTI

Project Title:

- Integration and Commissioning of a Prototype Federated Cloud for Scientific Workflows

Status:

- Three major work items:
  1. Virtual Infrastructure Automation and Provisioning,
  2. Interoperability and Federation of Cloud Resources,
  3. High-Throughput Fabric Virtualization.

Virtual Machines as Jobs

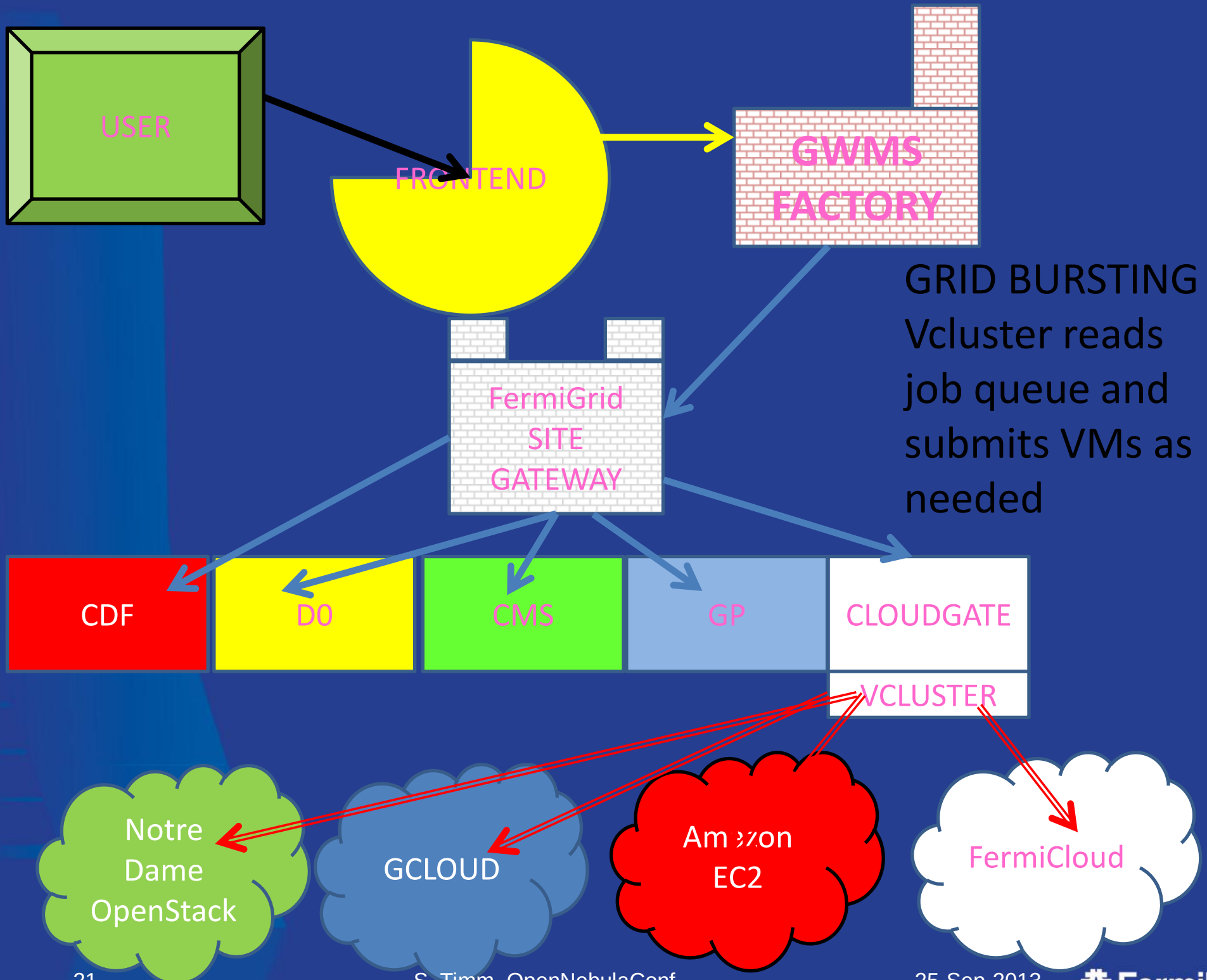
OpenNebula (and all other open-source IaaS stacks) provide an emulation of Amazon EC2.

HTCondor developers added code to their “Amazon EC2” universe to support the X.509-authenticated protocol.

Currently testing in bulk, up to 75 VM’s OK thus far:

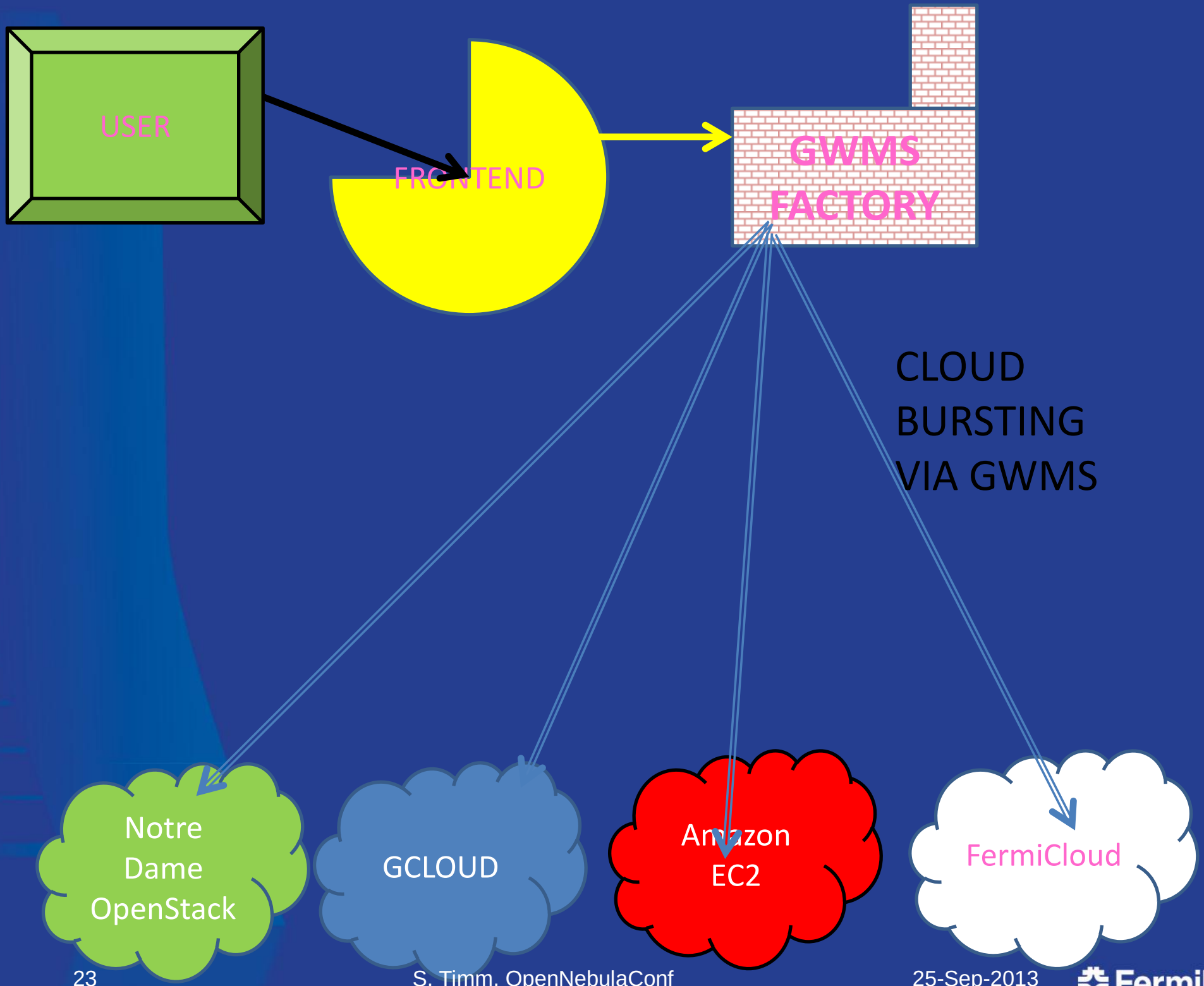
Goal to submit NOvA workflow to OpenNebula @ FermiCloud, OpenStack @ Notre Dame, and Amazon EC2.

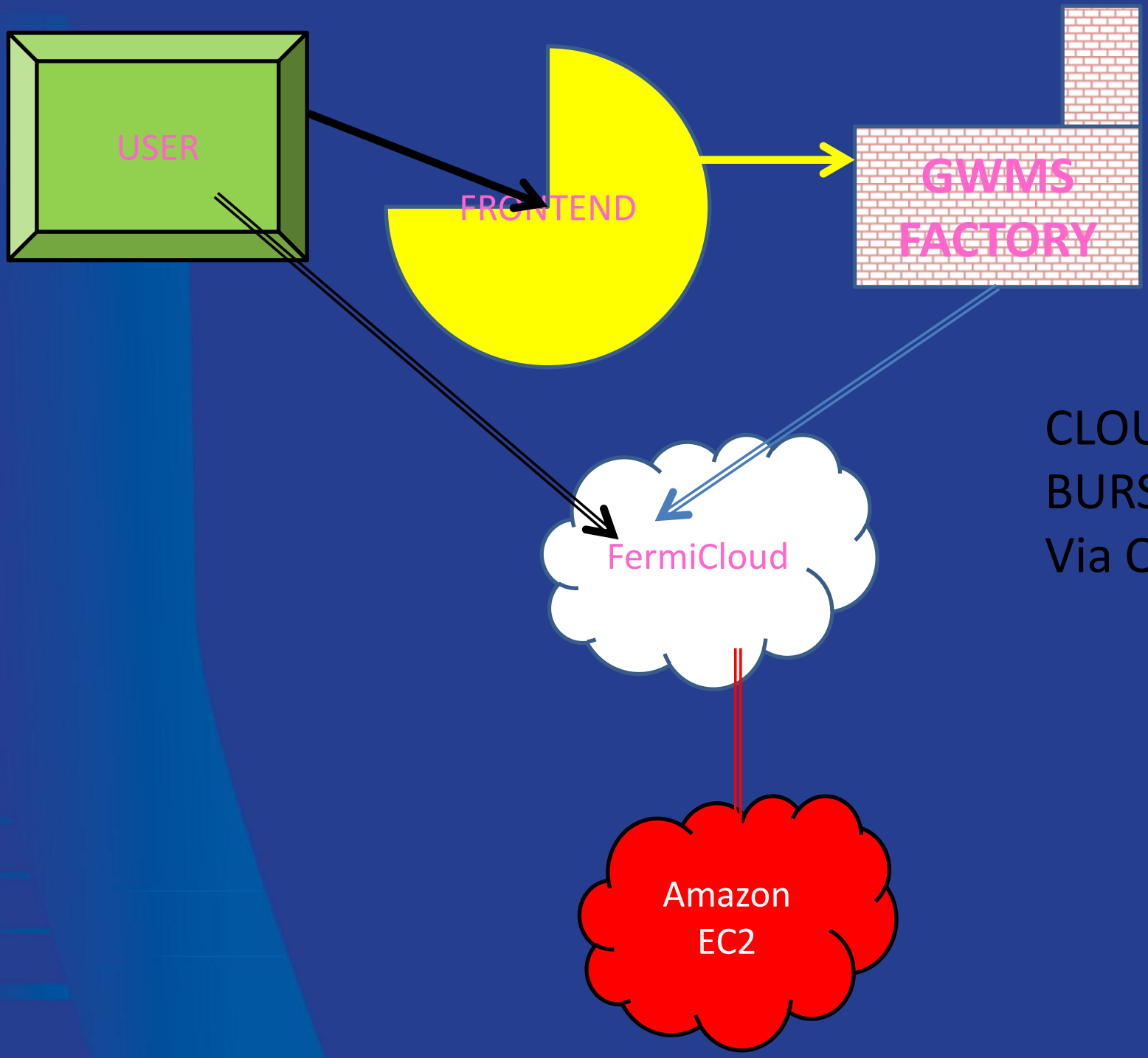
Smooth submission of many thousands of VM’s is key step to making the full infrastructure of a site into a science cloud.



GRID BURSTING
Vcluster reads job queue and submits VMs as needed

vCluster at SC2012


USER
FRONTEND
GWMS
FACTORY
CLOUD
BURSTING
VIA GWMS
Notre
Dame
OpenStack
GCLOUD
Amazon
EC2
FermiCloud



CLOUD
BURSTING
Via OpenNebula

True Idle VM Detection

In times of resource need, we want the ability to suspend or “shelve” idle VMs in order to free up resources for higher priority usage.

- This is especially important in the event of constrained resources (e.g. during building or network failure).

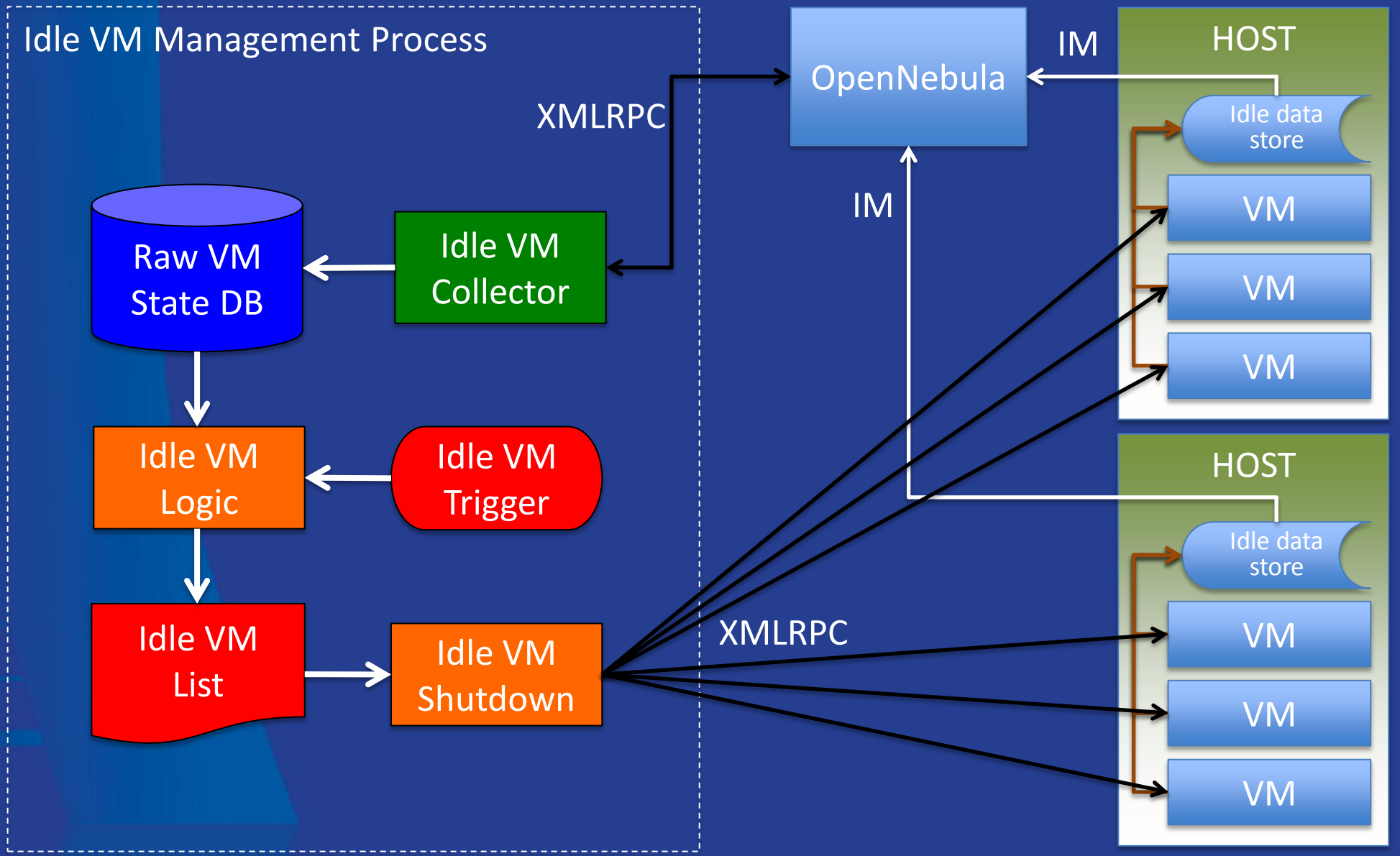
Shelving of “9x5” and “opportunistic” VMs allows us to use FermiCloud resources for Grid worker node VMs during nights and weekends

- This is part of the draft economic model.

Giovanni Franzini (an Italian co-op student) has written (extensible) code for an “Idle VM Probe” that can be used to detect idle virtual machines based on CPU, disk I/O and network I/O.

Nick Palombo, consultant, has written the communication system and the collector system to do rule-based actions based on the idle information.

Idle VM Information Flow



Interoperability and Federation

Driver:

- Global scientific collaborations such as LHC experiments will have to interoperate across facilities with heterogeneous cloud infrastructure.

European efforts:

- EGI Cloud Federation Task Force – several institutional clouds (OpenNebula, OpenStack, StratusLab).
- HelixNebula—Federation of commercial cloud providers

Our goals:

- Show proof of principle—Federation including FermiCloud + KISTI "G Cloud" + one or more commercial cloud providers + other research institution community clouds if possible.
- Participate in existing federations if possible.

Core Competency:

- FermiCloud project can contribute to these cloud federations given our expertise in X.509 Authentication and Authorization, and our long experience in grid federation

Virtual Image Formats

Different clouds have different virtual machine image formats:

- File system ++, Partition table, LVM volumes, Kernel?

We have identified the differences and written a comprehensive step by step user manual, soon to be public.

Interoperability/Compatibility of API's

Amazon EC2 API is not open source, it is a moving target that changes frequently.

Open-source emulations have various feature levels and accuracy of implementation:

- Compare and contrast OpenNebula, OpenStack, and commercial clouds,
- Identify lowest common denominator(s) that work on all.

VM Image Distribution

Investigate existing image marketplaces (HEPiX, U. of Victoria).

Investigate if we need an Amazon S3-like storage/distribution method for OS images,

- OpenNebula doesn't have one at present,
- A GridFTP "door" to the OpenNebula VM library is a possibility, this could be integrated with an automatic security scan workflow using the existing Fermilab NESSUS infrastructure.

High-Throughput Fabric Virtualization

Followed up earlier virtualized MPI work:

- Use it in real scientific workflows
- Now users can define a set of IB machines in OpenNebula on their own
- DAQ system simulation
- Large multicast activity
- Also experiments done with virtualized 10GBe on 100GBit WAN testbed.

Security

Main areas of cloud security development:

Secure Contextualization:

- Secrets such as X.509 service certificates and Kerberos keytabs are not stored in virtual machines (See following talk for more details).

X.509 Authentication/Authorization:

- X.509 Authentication written by T. Hesselroth, code submitted to and accepted by OpenNebula, publicly available since Jan-2012.

Security Policy:

- A security taskforce met and delivered a report to the Fermilab Computer Security Board, recommending the creation of a new Cloud Computing Environment, now in progress.

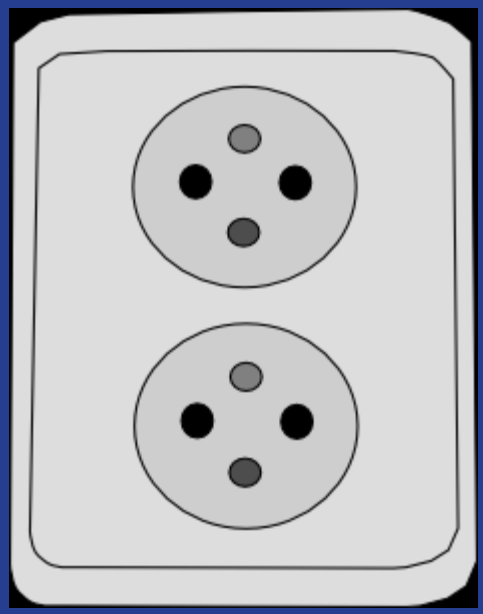
We also participated in the HEPiX Virtualisation Task Force,

- We respectfully disagree with the recommendations regarding VM endorsement.

Pluggable authentication

Some assembly required

Batteries not included



OpenNebula Authentication

OpenNebula came with “pluggable” authentication, but few plugins initially available.

OpenNebula 2.0 Web services by default used access key / secret key mechanism similar to Amazon EC2. No https available.

Four ways to access OpenNebula:

- Command line tools,
- Sunstone Web GUI,
- “ECONE” web service emulation of Amazon Restful (Query) API,
- OCCI web service.

FermiCloud project wrote X.509-based authentication plugins:

- Patches to OpenNebula to support this were developed at Fermilab and submitted back to the OpenNebula project in Fall 2011 (generally available in OpenNebula V3.2 onwards).
- X.509 plugins available for command line and for web services authentication.

X.509 Authentication—how it works

- Command line:
  - User creates a X.509-based token using “oneuser login” command
  - This makes a base64 hash of the user’s proxy and certificate chain, combined with a username:expiration date, signed with the user’s private key
- Web Services:
  - Web services daemon contacts OpenNebula XML-RPC core on the users’ behalf, using the host certificate to sign the authentication token.
  - Use Apache mod_proxy to pass the grid certificate DN to web services.
- Limitations:
  - With Web services, one DN can map to only one user.

Grid AuthZ Interoperability Protocol

- Use XACML 2.0 to specify
  - DN, CA, Hostname, CA, FQAN, FQAN signing entity, and more.
- Developed in 2007, has been used in Open Science Grid and other grids
- Java and C bindings available for client
  - Most commonly used C binding is LCMAPS
- Used to talk to GUMS, SAZ, others
- Allows one user to be part of different Virtual Organizations and have different groups and roles.
- For Cloud authorization we will configure GUMS to map back to individual user names, one per person
- Each personal account in OpenNebula created in advance.

“Authorization” in OpenNebula

- Note: OpenNebula has pluggable “Authorization” modules as well.
- These control Access ACL’s—namely which user can launch a virtual machine, create a network, store an image, etc.
- Not related to the grid-based notion of authorization at all.
- Instead we make our “Authorization” additions to the Authentication routines of OpenNebula

X.509 Authorization

- OpenNebula authorization plugins written in Ruby
- Use existing Grid routines to call to external GUMS and SAZ authorization servers
- Use Ruby-C binding to call C-based routines for LCMAPS or
- Use Ruby-Java bridge to call Java-based routines from Privilege proj.
- GUMS returns uid/gid, SAZ returns yes/no.
- Works with OpenNebula command line and non-interactive web services
- Much effort spent in trying to send user credentials with extended attributes into web browser
- Currently—ruby-java-bridge setup works for CLI
- For Sunstone we have shifted to have callout to VOMS done on server side.
- We are always interested in talking to anyone who is doing X.509 authentication in any cloud.

Reframing Cloud Discussion

Purpose of Infrastructure-as-a-service:

On demand only?

No—a whole new way to think about IT infrastructure both internal and external.

Cloud API is just a part of rethinking IT infrastructure for data-intensive science (and MIS).

Only as good as the hardware and software it's built on.

Network fabric, storage, and applications all crucial.

Buy or build?

Both! Will always need some in-house capacity.

Performance hit?

Most can be traced to badly written applications or misconfigured OS.

FermiCloud Project Summary - 1

Science is directly and indirectly benefiting from FermiCloud:

- CDF, D0, Intensity Frontier, Cosmic Frontier, CMS, ATLAS, Open Science Grid,…

FermiCloud operates at the forefront of delivering cloud computing capabilities to support scientific research:

- By starting small, developing a list of requirements, building on existing Grid knowledge and infrastructure to address those requirements, FermiCloud has managed to deliver a production class Infrastructure as a Service cloud computing capability that supports science at Fermilab.
- FermiCloud has provided FermiGrid with an infrastructure that has allowed us to test Grid middleware at production scale prior to deployment.
- The Open Science Grid software team used FermiCloud resources to support their RPM “refactoring” and is currently using it to support their ongoing middleware development/integration.

FermiCloud Project Summary

The FermiCloud collaboration with KISTI has leveraged the resources and expertise of both institutions to achieve significant benefits.

vCluster has demonstrated proof of principle Grid Bursting" using FermiCloud and Amazon EC2 resources.

Using SRIOV drivers on FermiCloud virtual machines, MPI performance has been demonstrated to be >96% of the native "bare metal" performance.

The future is mostly cloudy.

Acknowledgements

None of this work could have been accomplished without:

- The excellent support from other departments of the Fermilab Computing Sector – including Computing Facilities, Site Networking, and Logistics.
- The excellent collaboration with the open source communities – especially Scientific Linux and OpenNebula,
- As well as the excellent collaboration and contributions from KISTI.
- And talented summer students from Illinois Institute of Technology